|  |  |
| --- | --- |
|  | Hallo iedereen, Ik ben Dries Kennes |
|  | Ik heb als onderwerp voor project 1 gekozen om het klassiek spel Mastermind te digitaliseren.  De bediening van het spel gebeurt doormiddel van een smartphone vriendelijke web interface. Het spelbord word ook weergegeven doormiddel van RGB LEDS. |
|  | Om dit te realiseren heb ik gebruik gemaakt van het platform RIOT dat door meneer Pauwels en meneer Roggemans is ontworpen, ondertussen meer dan 10 jaar geleden.  Ik heb wel mijn eigen versie van de print gemaakt omdat het RIOT bord overbodige componenten heeft zoals RS232 levelshifters en een schakelende voeding. Ook kon ik zo ervaring opdoen met Altium en het ontwerpen van printplaten.  Het originele RIOT bord staat op de foto links, mijn versie rechts.  Het bord werk volgens dit principe: 1 minicomputer die op een DOS achtig real time operating system draait (de SC 12 in de oranje karder) kan via een gedeeld geheugen (in de blauwe kader) communiceren met een slimme IO processor (in de groene kader). |
|  | Omdat je voor een mastermind bord bijna 100 leds nodig hebt, heb ik ervoor gekozen WS2812 LEDs te gebruiken, aangezien die makkelijker in grote hoeveelheiden aan te sturen zijn dan bijvoorbeeld een matrix. Ze hebben namelijk maar 1 pin nodig. De driver is wel zeer tijdskritiek, maar aangezien ze veelgebruikt zijn in allerlei projecten, is er enorm veel informatie en hulp te vinden op het internet. Ook deze printplaten heb ik zelf ontworpen. Het gebruiken van losse LEDS op een breakout bord is natuurlijk enkel mogelijk omdat dit een prototype is, op een productiemodel zou dit systeem veel te veel manuren kosten. |
|  | Om het project gebruiksvriendelijker te maken heb ik ook een LCD module gebruikt. Die word gebruikt om het ip adress of status berichten weer te geven. Ook hier heb ik zelf een simpele printplaat voor getekend. (de rechtse op de foto) De print is ook voorzien van een kleine prototype area, omdat de backlight van de meeste LCD modules niet op dezelfde manier moet worden aangestuurd. |
|  | Omdat het RIOT bord gebruik maakt van 2 controllers, een beck sc12 minicomputer en een AVR Atmega128 als slimme IO chip, heb ik eigenlijk 2 programma's geschreven. Ze zijn allebei geschreven in C, met uitzondering van de WS2812 driver. Omdat die zo tijdskritiek is, is hij in inline ASM geschreven.  Het programma in de AVR is niet meer dan een IO interface. Het is enkel actief als de SC12 daar via interrupt het commando voor geeft.  Het programma in de SC12 bevat de web interface en regelt het spelverloop. Dit programma is het grootst, en het meest ingewikkeld. Het is wel gedocumenteerd, iets wat ik normaal wel al eens durf overslaan.  Het gebruiken van een ouder hardware (de SC12) heeft nadelen. De software om programma's te schrijven word niet meer ondersteund en ze is weinig informatie over te vinden online, maar ik ben er wel in geslaagd het bruikbaar te maken. Onder meer dankzij een korte handleiding op de Telescript FTP server. |
|  | Dit is hoe de software in de AVR werkt  Het hoofdprogramma doet niet meer dan het initialiseren van de nodige onderdelen en gaat dan in een eeuwige lus.  Als er vanuit de SC12 een interrupt word gestuurd, word er een bepaalde driver geactiveerd, die dan zijn data uit het gedeeld geheugen haalt.  Na het afhandelen van de interrupt gaat de controller weer naar wachten op de volgende interrupt. Een 2e interrupt tijdens een eerste word afgehandeld nadat de 1e klaar is, een 3e of meer gaat verloren. Dit is vooral belangrijk bij het updaten van de LCD, daarom dat er bij de SC12 LCD drives ook een delay is ingebouwd. |
|  | Dit is een deel van de flowchart van de SC12. Dit is het hoofdprogramma en de lcd update taak. In dit programma zijn uitbreide debug mogelijkheden voorzien, die via telnet of uart beschikbaar zijn.  De volledige flowchart van de webserver is te groot om goed weer te geven in deze presentatie, maar staat op de Trac en in het verslag.  Er zijn 5 pagina’s, home (welkom pagina), start (om een nieuw spel te beginnen), pickUsername (om een naam te kiezen), reset (om het spel te resetten na winnen of verliezen), maar de belangrijkste pagina is de play pagina, waar het spel word gespeelt. Op de play pagina worden ook de gokken uitgevoerd, en in de functie van die pagina word ook het LED scherm geupdate.  Omdat de webserver geen sockets ondersteund en niet snel genoeg is om van een aantal gebruikers elke x seconden een statusupdate te krijgen, update deze pagina niet als er een gok word gemaakt door iemand anders. Aangezien het de bedoeling is dat men het LED scherm kan zien vind ik dit geen prioritair probleem. |
|  | Ten slotte nog een foto van het eindresultaat. Er zijn 2 spelmogelijkheden: Allen tegen een of allen tegen de computer, als men speelt in ‘allen tegen een’ kan degene die de code heeft gekozen natuurlijk niet mee raden. |